Architecture Overview and Data Analysis Approach of the eTelematik ICT-System

Outline of System Requirements, Implementation Design, Field Test Results and Analysis approach

Steffen Späthe
Department of Computer Science
Friedrich Schiller University Jena
Jena, Germany
steffen.spaethe@uni-jena.de

Robert Büttner
Navimatix GmbH
Software Department
Jena, Germany
robert.buettner@navimatix.de

Wilhelm Rossak
Department of Computer Science
Friedrich Schiller University Jena
Jena, Germany
wilhelm.rossak@uni-jena.de

***Abstract*—Electrical vehicles are not only passenger cars but also commercial vehicles and, in particular, municipal vehicles. Their acceptance and usage depends primarily on everyday usability, aiming for a smart vehicle with intelligent energy and range supervision as well as driver support. In our funded research project eTelematik, we conceptualized, implemented and proved an Information and Communication Technology (ICT) System with directly connected vehicle components, driver interface and back end applications as well as an analytical evaluation process for our prediction model. In order to expand the usage of electric vehicles, we predict energy consumption of complex work task sets and guide vehicle drivers while driving.**

Keywords - Municipal vehicles; ICT-support for fully electric vehicles; range prediction; mobile client; in-car module; trajectory; Dynamic Time Warping.

I. Introduction

This paper is an extended version of a paper that was presented at the Forth International Conference on Smart Systems, Devices and Technology, SMART 2015 [1].

Worldwide electrical vehicles are seen as the future of mobility. The primary focus in this vision is mainly on private cars [2]. However, commercially used vehicles have a much better starting point for electrification. Based on prescheduled tasks and daily high usage work, the capability of commercial vehicles can be predicted. At the current state of development commercially used, fully electrical vehicles are not able to fulfill a full day's work without recharging. Therefore, hybrid vehicle concepts are developed and currently in advanced prototype state. A special class of commercial vehicles are municipal vehicles. These universal vehicles can be used with different setups and add-on structural parts in various scenarios.

In our research project eTelematik, we developed a system based on Information and Communication Technologies (ICT), which supports daily commercial usage of electrical municipal vehicles and allows for new usage scenarios with hybrid vehicles.

The project eTelematik was a federal funded research project during 2012 and 2014. The consortium included four main partners [3]:

1. EPSa GmbH: industry, electronics and communication devices
2. Navimatix GmbH: mobile and server applications
3. Friedrich Schiller University Jena: research, distributed software systems, range estimation
4. HAKO GmbH Werk Walterhausen: electrical municipal vehicles

The paper presented here, focuses on summarizing the project and its general accomplished results as well as giving a detailed overview of the analysis approach used for the evaluation of the produced data within the project.

The remainder of the paper is organized as follows: In Section II we will provide an overview of the project's overall ICT architecture including the main challenges of our distributed system. From there we will highlight the usage of collected data inside the vehicle and on back end systems. In Section III the analysis process for the evaluation of work task sets will be described and in Section IV some information and findings of the project's long-term field test will be presented. We close with a short review of goal reaching in Section V.

II. The eTelematik project

The main focus of the project was the creation of a complete ICT infrastructure to enable an improved usage of electrical municipal vehicles.

Our main requirements for this system were

a) to gather data from mobile electrical vehicles and store them in a central universal database,
b) to interpret gathered data in order to evaluate the influence of various parameters on energy consumption during the fulfillment of certain work tasks with required work equipment,
c) to adjust the internal energy consumption and range prediction model with computed factors of influence and
d) to support the driver with information about estimated and real energy consumption of current and scheduled work tasks, irrespective of the status of the connection to the central server.

Excluded from the project focus was the development of a new work force management or fleet management/ optimization system. Thus, all required business data had to be provided from an external fleet management system via designated service interfaces.

Based on these requirements, we developed our system as schematically shown in Figure 1.

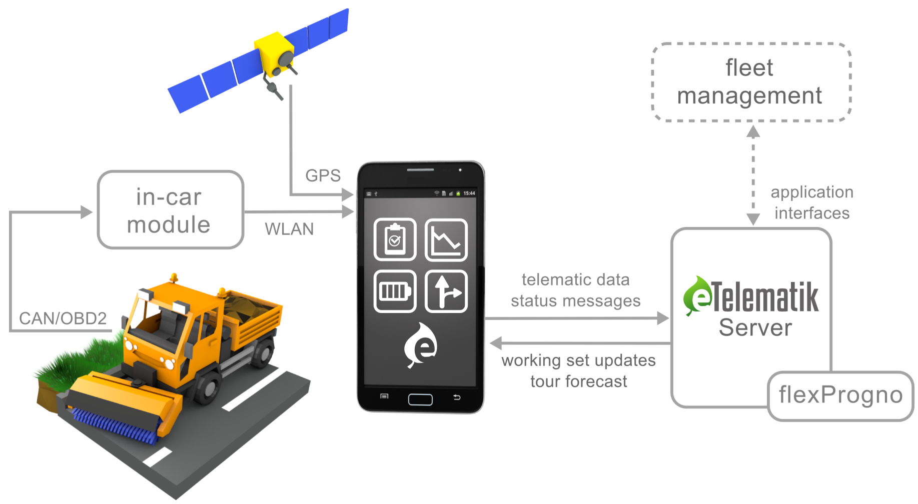

Fig. 1. eTelematik system architecture overview (adapted figure, based on original by Johannes Kretzschmar, University Jena)

The eTelematik solution consists of a communication hardware ("in-car module"), a mobile application ("mobile client") and a central server ("central instance") with a prediction model ("flexProgno").

Externally computed work task sets are evaluated in regard to their practicability in our central instance *eTelematik Server*. We use our energy consumption and range prediction model *flexProgno* to estimate the power consumption for every single part of the given work task set. While power consumption depends on various parameters like vehicle model, payload, environmental temperature (as already shown in [4]), we need to know more about the work task, required add-on structural parts and settings of them. Moreover, we require knowledge about the concrete routing and their elevation profile between different work task places. We use the commercial available route calculation service and map height services of project partner Navimatix GmbH to gather this data.

If a working set is estimated as achievable, the assigned driver gets this set shown on his mobile client.

Inside the vehicle the communication hardware, developed by EPSa GmbH, collects vehicle specific data in real-time, aggregates and sends them to the mobile client. Communication between the communication hardware and the vehicle is realized by Controller Area Network (CAN) connections. The mobile client, developed by Navimatix GmbH, is an Android application running on established consumer devices. The mobile client informs the driver about the actual operating status of the vehicle, the current status prediction based on the assigned working task set and the probability of fulfillment of this set. All collected vehicle data combined with sensor data from the mobile phone are transferred to and stored at the central instance.

Figure 2 shows the internal conceptual system design of the mobile client application. The mobile client is subdivided into a user interface related part and some background services. While the data storing modules are responsible for realizing business logic, which is used by the UI module, the ICM Communicator Service takes care of establishing a connection to the in-car module and keeping it alive. We use plain TCP socket connections at this communication channel to minimize transport size and delay overhead. The eTele App Communicator Service realizes the reliable communication to the central instance. All other services, background as well as UI-related, use this service to communicate to and receive data from the central instance. At this channel, we use HTTP as transport layer. Since our JBoss Application Server based central instance is realized by using Java Servlets and Enterprise Java Beans, HTTP is a natural choice. As payload, we used data objects with an own implemented key-value based object serialization which represents our business data.

In summary, our system has to handle the following data from central instance to vehicle:

- master data of vehicles and drivers
- general and vehicle specific configuration setting for communication between in-car module and mobile client
- current work task sets depending on logged in driver

From vehicles to central instance we send:

- updates of work task status
- vehicle's positions and velocity
- electrical vehicle specific measurements

The electrical vehicle specific measurements and especially their representation on in-car communication buses vary between vehicle manufacturers and even between vehicle types of one manufacturer. Within our project consortium, we are able to gather and transfer the following electrical vehicle specific measurements:

- state of charge
- primary battery voltage
- current in high voltage circuit
- connection state, settings and power of battery recharger
- state and settings of range extender (if applicable)

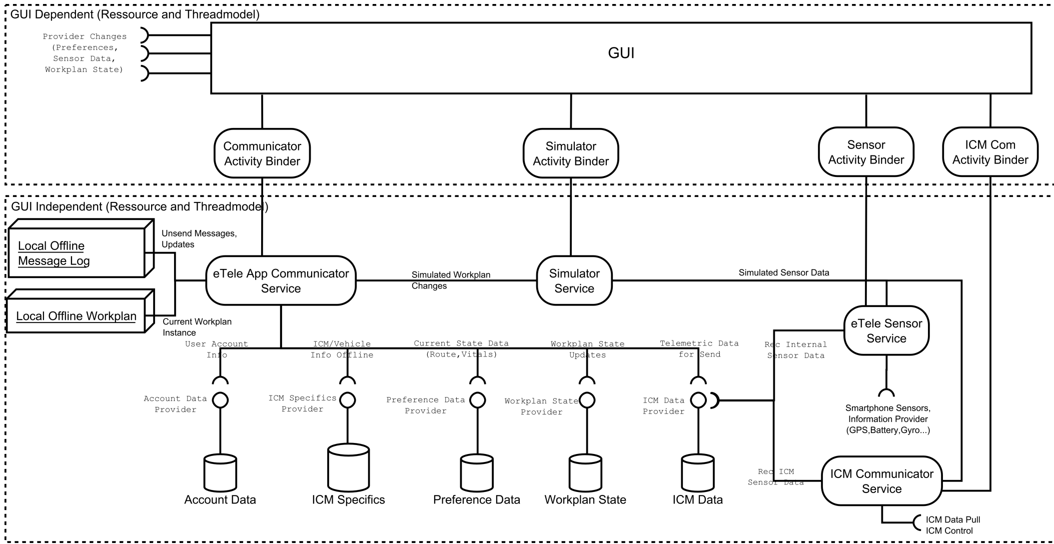

Fig. 2. eTelematik mobile clients internal module overview

This data is used in different situations. The data supports the driver inside the vehicle in driving between work task places of action and while task fulfillment. On server side, we use recorded data in different analyses. Some are shown in Section III of this paper.

Inside the vehicle, we are able to realize "enhanced fore-sighted driving". Since we know, based on the scheduled work task sets, which route has to be driven and what kind of working task has to be accomplished, we are able to predict if this planning is still valid. Usually, only average statistics about energy consumption per kilometer are available to the vehicle and the driver. We know the exact route to drive as well as the required settings of add-on components. Thus, we are able to predict the required energy consumption on a much more detailed basis. This advanced, detailed knowledge allows us to warn the driver that he will not reach his destination, even if the average statistics would tell him so. Alternatively, we can relax him in situations where average statistics would show a much too low range, for example, when the planned route has many downhill sections. Furthermore, we can delay the usage of the range extender in hybrid cars when it would be triggered by the vehicle's management system, because we know when the user's preferred charging stations are in reach.

By doing so, we are able to optimize the battery usage and extend the usability of electrical vehicles.

In vehicles with range extender, we can optimize the point in time for recharging. In certain situations, work tasks have to be fulfilled without any avoidable emission, e.g., noise or exhaust. If recharging is only controlled by battery state of charge, it could happen that the driving to the workplace is realized fully out of battery and that the recharging has to be started at the workplace. With our knowledge about the complete work task set and desired or required restrictions in work task fulfillment, we can foresee and avoid such situations.

On server side, we use recorded data of the vehicle in different scenarios.

A long-term use case, which is very important to vehicle manufactures as well as to the vehicle owner, is predictive maintenance. With data mining techniques, we are able to detect deviations in characteristic gradients long before the vehicle breaks down. This is of particular interest to our project partners due to the lack of long-term experience with the completely new designed power train and the used battery system.

In addition, for the first time, this process enables insight into exhaustive detailed real world usage records of these vehicles. This information is very helpful to vehicle manufactures for further improvements and new developments.

A short-term use case is monitoring the overall resource consumption for certain work tasks. It is not possible to date the direct assignment of fuel consumption to single work tasks. Since we record work task state changes as well as energy consumption parameters continuously, we are able to match them.

Inside our project's system, we also process recorded data for intrasystem usage. The main task is to adjust and improve our energy consumption and range prediction model flexProgno. Our model is based on assumptions, e.g., required energy stays equal if all influencing parameters do not change or stay very close to situations before. Initially, we did not have many vehicle specific data. By processing recorded data, the vehicle specific parameter set gets more accurate over time. The basic approach of our model is shown in [5].

Energy consumption does not only depend on vehicle or work task parameters, but also on driver characteristics. Hence, it is important to include the driver's start and stop behavior in energy consumption prediction. Since these parameters cannot be measured beforehand, they need to be determined from the recorded data.

III. Location-based analysis for work task sets

A working task set is defined as a round trip with several work task places.

Figure 3 shows an abstract work task set. A driver starts the trip at a central point ("start"), driving over to his work task places, e.g., A, B and C, where the driver will then fulfill the work tasks. After finishing all tasks in the given order, the driver will then drive back to the central point ("end"), which is not necessarily the same as the starting point.

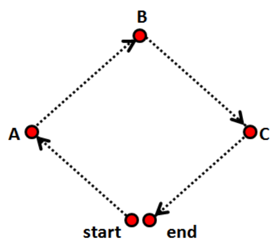

Fig. 3. Work task set definition as round trip with work task places A, B and C

To verify our assumptions about a certain work task set with its predicted routes between task places, it is worth to compare recorded data with the predicted data or even with other recorded data of the same task set. Furthermore, the comparison of different trip records of the same route at specific spatial scenarios, e.g., uphill, downhill, highway or city traffic, can help us to understand the behavior of drivers and vehicles. Hence, enables us to evaluate and adjust our prediction model.

While doing so, the data can be examined from different perspectives and all of them may raise different questions. For example:

A) *Geographically*, i.e., "Is the driven route equal to the predicted route?"
B) *Energy consumption*, i.e., "Is the predicted consumption close to the real consumption?" or "Which driver was saving most energy on the same route as others and why?"
C) *Trip time*, i.e., "Is the predicted estimated trip time close to real trip times?"

For the analysis, we use the recorded trajectory data of each municipal electric vehicles using our system architecture. As above mentioned, we want to analyze them at specific location-based scenarios. Thus, we always need to specify a spatial reference track which defines the road segment that is going to be analyzed. Both, trajectory and reference track, are defined as follows in definitions 1 and 2:

Definition 1. *A trajectory* $T = \{p_1, p_2, \ldots, p_N\}$ *is a finite sequence of points. Each point* $p_i = \{ts, pos, o\}$ *consists of a timestamp (ts) and a geo-position* $pos = \{lon, lat, alt\}$ *with longitude (lon), latitude (lat) and altitude (alt). An optional set of attributes (o) with additional measurements for each spatio-temporal point can be defined. All points are in temporal order* $p_1.ts < p_2.ts < \ldots < p_N.ts$.

Definition 2. *A reference track* $rT = \{p_1, p_2, \ldots, p_N\}$ *is a finite ordered sequence of points. Each point* $p_i = \{pos\}$ *consists of a geo-position* $pos = \{lon, lat, alt\}$ *with longitude (lon), latitude (lat) and altitude (alt).*

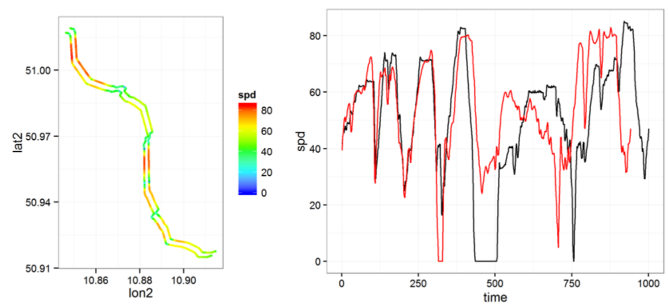

Fig. 4. Two similar recorded tracks, side by side, with colored speed (left); speed vs. time graph of the same two tracks (right)

However, the representation form of a temporal trajectory is very unsuitable for a comparison on a local basis. In fact, looking at the left graph of Figure 4, the human brain might be able to perform such a comparison given the appropriate visual representation. Unfortunately, a computer using algorithms cannot do that, due to temporal shifts or distortion and the consequential difference in length, as shown in the right graph of Figure 4. Thus, it is necessary to synchronize the data at a geographical-spatial basis to be also able to analyze it automatically through algorithms.

The synchronization is realized using the *Dynamic Time Warping* (DTW) Algorithm. The DTW Algorithm is well known in the area of time series alignment and clustering. One of its first applications was speech recognition back in the 70s and since then it is also used in handwriting and gestures recognition, to only name a few [6], [7]. To overcome the limitations of shifts in time series, it generates a warping path which represents an optimal alignment between the two, not necessarily equally long, time series, as shown in Figure 5.

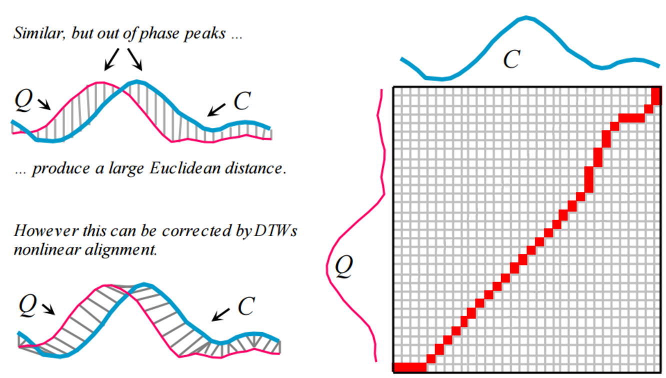

Fig. 5. Point correspondence when two similar time series contain local time shifting using Euclidean distance (upper left); using DTW (lower left); Search for an optimal alignment or warping path (red squares) within the distance matrix between the same two time series (right) Image source: [8]

A. Methodology

Our analysis process can be divided into the three subprocesses *(i)* Data preparation, *(ii)* Identification of analyzable segments and *(iii)* Spatial synchronization:

(i) Data preparation: Before it is possible to perform a data analysis, the data needs to be structured in a way that consistent processing can be guaranteed.

As already mentioned in Section II, the electrical vehicle specific measurements and their representation can vary between vehicle types, but also the time intervals in which the various measurements are collected within the vehicle can differ. Taken all that into account, it is necessary to convert the data into an homogeneous data scheme. To overcome the measure interval differences, the data will be embedded into a fixed time interval, e.g., every n seconds. Emerging temporal gaps will eventually be filled using a linear interpolation.

Due to the possible occurrence of errors during data collection, it is also necessary to perform a data cleansing. Especially when using GPS, it is not uncommon to receive erroneous or inaccurate GPS positions. In order to correct these position data, we use a map-matching algorithm which brings off-track positions back to their corresponding road element using the map service of project partner Navimatix GmbH.

After homogenization and data cleansing, we also perform a data enrichment to combine our vehicle specific measurements with more general information, e.g., traffic, temperature and weather or even more accurate altitude information, to improve our data even more.

The result of this subprocess is an homogenized, revised trajectory dataset $QT = \{T_1, T_2, \ldots, T_N\}$, which now can be used as a basis for the analyzing process.

(ii) Identification of analyzable segments: Given a preselected reference track rT, which could contain interesting sections for an analysis task, we need to determine all common sub-trajectories between rT and the dataset QT in order to find all track segments $RT' = rT \cap QT$ that can possibly be covered by the dataset. Those segments need to be clustered into distinct groups and will be afterwards presented to an expert who will finally choose one segment, which will serve as the selected reference track $rT' \in RT'$ for the further detailed analysis process.

(iii) Spatial synchronization: Since a reference track rT' has been selected by the expert, the spatial synchronization to enable the location-based analysis can be performed.

To do that, we need to find all common sub-trajectories $CS = rT' \cap QT$ that cover the selected reference track entirely. Afterwards, the spatial synchronization takes place between rT' and every common sub-trajectory $cs \in CS$ and results in synchronized common sub-trajectories cs_{sync} with the same length as the selected reference track. This implies that, with the definition of the preselected reference track rT, it is possible to control the level of detail of the synchronization outcome, i.e., sample points and distance between them.

The trajectories cs_{sync} are now equally long and it is guaranteed that the i-th element of each trajectory refers to the same spatial position. Hence, can be compared with each other using algorithms.

B. Algorithms

To reflect the spatial synchronization process two key-algorithms have to be implemented: *(i)* Determining common sub-trajectories and *(ii)* Spatial synchronization of common sub-trajectories.

(i) Determining common sub-trajectories: This algorithm determines common subsegments between two location-based data series, in our case reference track and trajectory, where local distances of corresponding points are within a tolerated distance r. The DTW algorithm is used to determine the corresponding points. For this, a pair-wise local distance matrix $D(rT, T) \in \mathbb{R}^{M \times N}$ is built between all positions of the reference track rT with M elements and the trajectory T with N elements. The distance between two positions will be calculated using great-circle distance calculation, i.e., a low-costed matrix, with a minimum of zero, will represent more geographic similarity than a high-costed matrix.

Based on matrix D the DTW algorithm calculates an alignment path, i.e., warping path, which runs through the low cost areas of the local distance matrix. It represents a complete assignment of all indices between both data series, starting with the first and ending with the last indices of both, to guarantee that every index is used at least once. The indices-pairs of the warping path are by default in a monotonically increasing order with a maximum step-size of 1.

Afterwards, the local distances of the warping path will be analyzed. Here, all index-pairs with a lesser local distance than the predefined tolerance radius r will be determined. These pairs are describing geographical common points CP between rT and T. Multiple consecutive common points can form a common sub-trajectory. To avoid large gaps between consecutive points, it is necessary to define a tolerance distance. The gap distance g represents the maximum distance two subsequent common points can be apart from each other to be recognized as "real consecutive" and, hence, forming a common sub-trajectory cs.

Figure 6 shows the local distances of the warping path's index-pairs. The straight colored line at the bottom side represents the tolerance radius for determining common points. Furthermore, aside from a common sub-trajectory, a gap between common points as well as a case of a possible crossing between the two trajectories is highlighted.

(ii) Spatial synchronization of common sub-trajectories: To make the local points of a common sub-trajectories $CS = \{cs_1, cs_2, \ldots, cs_N\}$ locally comparable with each other, the length of both data series needs to be equalized. Therefore, the points of the reference track will serve as spatial reference points. Each data series needs to be realigned to match the length of the reference track in order to return data for each

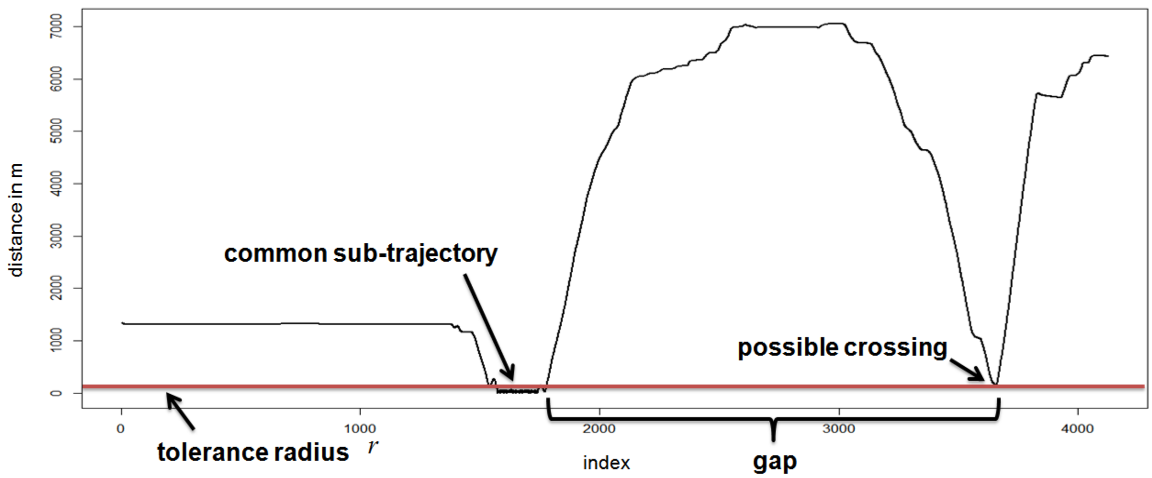

Fig. 6. Interpretation of the warping path's local distances in order to determine common sub-trajectories

spatial reference point. The alignment to realize this is already given by the warping path. However, before it can be used, multiple assignments of indices within the warping path need to be handled. They occur to compensate differences in length between the time series. The following cases of assignments are possible and need the given action:

- **normal case:** Exactly one index of cs is assigned to exactly one index of rT. Nothing needs to be done.
- **reduction case:** Multiple indices of cs are assigned to exactly one index of rT. Here, the multiple assigned points of cs need to be aggregated into one point, as shown in Figure 7.
- **extension case:** Exactly one index of cs is assigned to multiple indices of rT. This case needs to be handled very carefully to preserve the reference point count. Hence, an aggregation cannot be done. Instead the point of cs needs to be duplicated until every point of rT finds exactly one match, as also shown in Figure 7.

The result of the synchronization is a trajectory cs_{sync} which is exactly as long as the reference track rT' and can now be locally compared, i.e., by location, to other trajectories cs_{sync} synchronized on the same reference track.

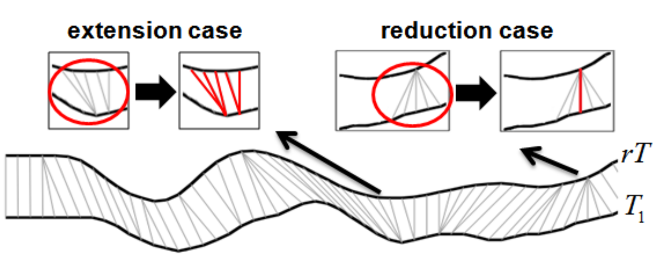

Fig. 7. Approaches of solving different multiple assignments problems within the warping path using duplication and aggregation (in this case mean calculation).

IV. The Project's Field Test

As we developed our system from scratch, we designed a long running field test. In this section, the test will be reviewed from two different perspectives. The technical perspective reviews the overall functionality of the system and whether all components are working well together. The analytical perspective is concerned with the resulting data produced by the system and the analytical potential of them.

A. *Technical perspective*

We built up a complete system installation to validate the system's long term stability, data transfer reliability especially in areas with unreliable mobile network connection and to validate and harden our prediction model.

In our build up test, we installed our system components in five electrical vehicles. These vehicles were used on a regular daily basis. In terms of the test, the following findings are worth mentioning:

A) Our system setup is running very stable over all components. During the development, there were some doubts about wireless local area network (WLAN) communication between in-car module and mobile client. However, we did not register any significant disturbance in this communication channel. All relevant data provided by the electrical vehicles in the field test were recorded by the in-car module and were transferred properly to the mobile client.

B) We succeeded in establishing a robust communication between mobile client and central instance. Even in our test region where mobile network coverage is very patchy, we had no data loss.

C) Synchronization of master data as well as measurement data between mobile client and central instance is working very solid, even if network connection gets lost while transfer. Thus, the required offline capability of the mobile client is achieved.

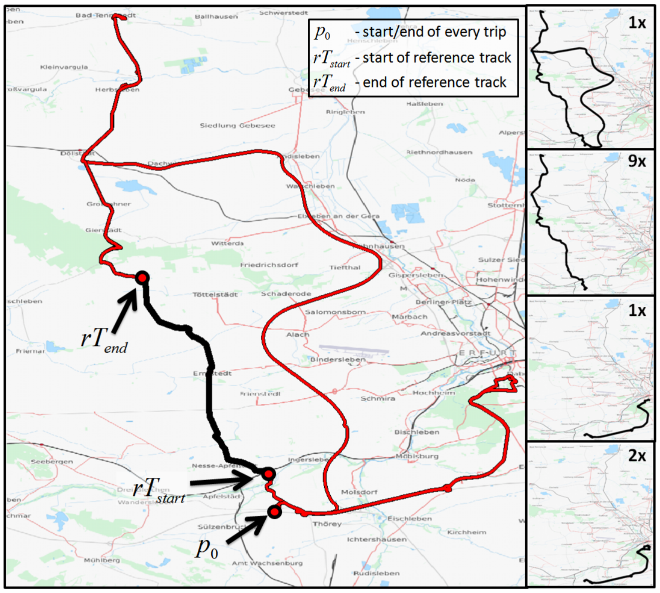

Fig. 8. Recorded subset of the road network with highlighted reference track and start/end point of every trip (left); trip types and their occurrences (right)

TABLE I. A part of the synchronized speed data of Figure 9

rT index	rT lon	rT lat	track1.spd	track2.spd
1000	10.88224	50.94711	37.0	17.3
1001	10.88224	50.94716	28.3	14.8
1002	10.88224	50.94720	28.3	14.8
1003	10.88224	50.94724	28.3	13.3
1004	10.88223	50.94729	20.5	8.9
1005	10.88223	50.94733	20.5	6.3

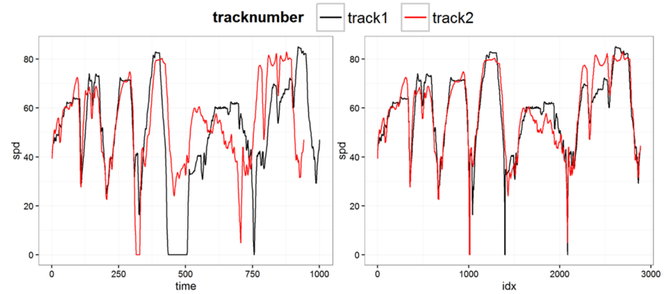

Fig. 9. Speed profile of two different trips on the same road segment in asynchronous representation (left); synchronized (right)

B. Analytical perspective

During the field test, the electrical vehicle specific measurements, mentioned in Section II, as well as GPS positions are collected and stored for each trip together with their corresponding timestamps per measurement, and hence forming trajectories. We are using the programming language R for our location-based analysis process to evaluate the consistence of the recorded data and our assumption about working task sets and the behavior of electrical vehicles in general.

For demonstration purposes, we examine an isolated and complete subset of the whole road network produced during the field test. Figure 8 shows the subset which was recorded in the area of Erfurt, Germany. It consists of 13 single trips which were all driven by the same car at different days. Each trip starts and ends at the same position p_0. The distinct tracks with the number of times they were used for the trips within the recorded data are listed at the right side of the figure. For the analysis, we choose the reference track highlighted in black and defined by the starting point rT_{start} and ending point rT_{end}.

During data preparation the recorded trip data is converted into a common data scheme and our map matching algorithm corrects positions that are off the road, which is crucial for the detection of common sub-trajectories. Additionally, the data is enriched with altitude data in order to complete the 3D position tuple and is now prepared for further analysis.

During the determining of common sub-trajectories between the reference track rT and the recorded trip data QT, ten common segments were found. All ten segments are fully covering the reference track. As Figure 8 indicates, this is expected, since this is equal to the number of trips sharing visually the same route as the reference track.

On these ten common segments, the spatial synchronization process is performed. To visualize the results of the process, Figure 9 is showing the speed profile of two of the ten common segments in an asynchronous state before and in a spatial synchronized state after the process. It is worth pointing out that the x-axis, which was representing time beforehand, has changed into the spatial dimension (namely “idx” for index) after the synchronization. The index scale is directly linked to the numbered elements of the reference track and their positions. A part of the data of Figure 9 together with the corresponding position data from the reference track is shown in Table I and proofs that the data can now be compared in a local dimension.

The representation in the local dimension opens up for new analysis perspectives. In order to see how different attributes depend on each other, we can now for example, analyze them not only over the course of one trip but across multiple trips with an identical route.

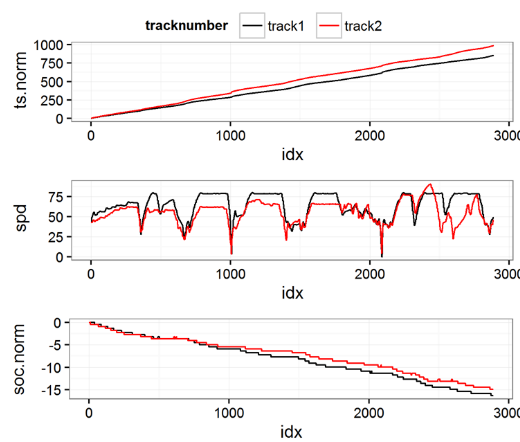

Fig. 10. Analysis of multiple attributes over two or more common segments

Figure 10 shows a possible visualization of such an analysis. Here, the three attributes *ts.norm* (trip duration), *spd* (speed) and *soc.norm* (state of charge loss) are presented. Note that, even though we left the time dimension due to the spatial synchronization, time data can still be restored as an analyzable attribute by storing the difference in time between two local positions as an attribute. The trip duration graph shows the duration of each trip in seconds. The state of charge graph allows us to identify the energy consumption on each trip. The speed graph shows the speed used at each position. It is noticeable that the difference in speed used during the different trip is responsible for the different trip durations that emerged. However, it could also be responsible for the difference in energy consumption, given our assumption that the speed, or better, the acceleration process has a significant impact on the energy consumption. Other attributes such as weather or altitude data could also show dependencies and could easily be added to the set of analyzable attributes for even more insight.

The synchronized data can also be aggregated on a local level. This can be used to create an energy consumption profile for a route with a mean energy consumption difference between positions of all trips. This could then be used to evaluate our predicted energy consumption model for a given route. Figure 11 shows the energy consumption profile for the reference track produced during the field test visualized by putting it on a map with a color indication for the mean energy consumption.

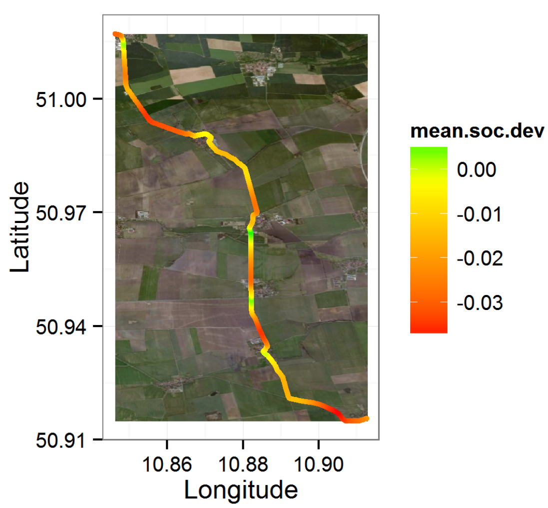

Fig. 11. Map of the reference track with mean energy consumption difference of all trip data of the route as color indication

During the analysis of the field test data, we noticed that the attribute *state of charge* is not satisfying our needs for range prediction evaluation purposes. The state of charge represents a percentage of the energy capacity in a battery. However, this maximum capacity can decrease due to extreme temperature or bad health state. The state of charge is calculated in relation to the current capacity. Without the current capacity measured in our system, it is needless to say that the state of charge cannot give us any information about the actual energy that is drained from the battery. This information is crucial for the comparison with our prediction model. Hence, in the future, instead of measuring the energy consumption as a relative value, we need to measure real energy values in kilowatt.

V. Summary and Future Work

Based on the evaluation of our long-term field test, we can state that we achieved our primary goals.

From a technical point of view, the overall data recording is satisfying, data transfer reliability is sufficient and offline capability for mobile client is achieved.

Therefore, we can determine that our selected system design and implementation are adequate to meet our overall requirements. However, we have to reassess our selection of mobile phone as primary communication channel. We deployed mass market mobile phones in the field test. So far we did not have substantial failures. Nevertheless, based on other tests we expect thermal problems in very cold and warm to hot situations. These problems will become more serious when running more applications and parallel tasks on the mobile phone's hardware.

Accordingly, the partitioning between in-car module and mobile client has to be reviewed very carefully. An alternative approach could be to transfer all permanent running processes of data collection and aggregation to the fixed-powered in-car module. This could as well include data transfer from and to the central instance. This process should be realized in a proxy-like way to keep this functionality transparent to the mobile client. The main function of the mobile phone still has to be the communication to the driver of the vehicle. This includes the exchange of information about working sets, as well as electrical vehicle state of charge, and driving instructions, to reach optimal range and energy usage.

A disadvantage within this alternative approach is the limited updatability and extensibility of the in-car module. The software for the in-car module has to be written system-specific and very closely fitted to the underlying hardware. Due to the rapid development of embedded Linux systems and their possibilities, we see now the option to overcome the above mentioned drawback by implementing our software on a Linux-based embedded system together with a scripting language to build a generic base system that is easily adaptable for the use on a specific underlying hardware.

From the analytical point of view, a process was introduced to synchronize spatial time series on a spatial-geographical basis. Although this process has to be tested thoroughly in the future, the evaluation of the field test shows that the overall functionality is working. Its provided functionality is helpful, as it allows us to evaluate our whole prediction model and the recorded data produced by our system, which would otherwise not be possible.

Thanks to the field test data evaluation, it is revealed that there is a lack of significance in some of our data attributes (namely state of charge), which provoked us to reconsider the electrical vehicle specific measurements that we are recording in general. However, at the time the field test took place, no other data was available for our system to measure. This has changed as the project proceeded and to date a lot more measurements can and will be recorded with our system. Hence, we can easily measure the required data, e.g., real energy consumption in kilowatt.

Until now, the analysis process has to be initiated manually on a specified reference track. In the future, the process could also be automatized to directly adjust our prediction model if multiple trips are showing a pattern of high deviations.

Still many questions and tasks are left open. Our work will be continued, partly in cooperation with the federal funded project called "Smart City Logistik Erfurt" (SCL) [9].

In SCL, we address aspects of inner city freight logistic processes with full electric vehicles. The logistic partners of SCL intend to deploy available medium sized electrical vehicles into their business as freight transporters for the last mile, from the city's perimeter to the final destination. The project's focus is on ICT support to optimize vehicle's utilization and integration in existing fleets and processes.

Therefore, we have to adapt our in-car module to the selected vehicle models. The driver assistance mobile application needs adjustments to meet the specific needs in delivery logistic applications. Our range prediction has to be adjusted to the new domain as we have differing influence factors like weight or specific vehicles accessories. In SCL, we will not only validate existing working sets. Implementation of route calculation and tour optimization with electrical vehicle's additional restrictions will be an important task. Overall, we have to improve usability and user experience in our driver assistance application as well as in the back-end system's user interface, which was not the focus of eTelematik, but is undoubtedly important to bring our research and development into real world applications.

VI. Acknowledgment

We would like to thank all members and partners of eTelematik research project, especially S. Gebhardt from HAKO GmbH for long lessons on battery chemistry and vehicle driving dynamics.

VII. Funding sources

This project was funded by the Federal Ministry for Economic Affairs and Energy (KF3012602).

References

[1] S. Späthe and W. Rossak, "etelematik: Ict-system for optimial usage of municipal electric vehicles," in *SMART 2015, The Fourth International Conference on Smart Systems, Devices and Technologies*, pp. 80–83.

[2] S. Lukic and Z. Pantic, "Cutting the cord: Static and dynamic inductive wireless charging of electric vehicles," *Electrification Magazine, IEEE*, vol. 1, no. 1, pp. 57–64, Sept 2013.

[3] etelematik project website. [Online]. Available: http://www.etelematik.de/ [Accessed Jun. 6, 2016].

[4] V. Schau *et al.*, "Smartcitylogistik (scl) erfurt: Deriving the main factors that influence vehicle range during short-distance freight transport when using fully electric vehicles," in *LBAS Conference on Location Based Applications and Services*, pp. 101–108.

[5] J. Kretzschmar, F. Geyer, S. Späthe, and W. Rossak, "etelematik: Prognose und ausfhrungsberwachung elektrifizierter kommunaler nutzfahrzeuge," in *LBAS Conference on Location Based Applications and Services*, pp. 119–125.

[6] P. Senin, "Dynamic time warping algorithm review," [Online]. Available: http://seninp.github.io/assets/pubs/senin_dtw_litreview_2008.pdf [Accessed Jun. 6, 2016].

[7] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *First SIAM International Conference on Data Mining (SDM)*, 2001.

[8] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 262–270, [Online]. Available: http://doi.acm.org/10.1145/2339530.2339576 [Accessed Jun. 6, 2016].

[9] Smart city logistik project website. [Online]. Available: http://www.smartcitylogistik.de/ [Accessed Jun. 6, 2016].